# Shift, scaling and derivative properties for the discrete cosine transform

Robert Reeves [a,*], Kurt Kubik [b]

[a] *School of Mathematical Sciences, Queensland University of Technology, GPO Box 2434, Brisbane, Q, 4001, Australia*

[b] *School of Information Technology and Electrical Engineering, University of Queensland, Brisbane, Q, 4072, Australia*

**Abstract**

A set of DCT domain properties for shifting and scaling by real amounts, and taking linear operations such as differentiation is described. The DCT coefficients of a sampled signal are subjected to a linear transform, which returns the DCT coefficients of the shifted, scaled and/or differentiated signal. The properties are derived by considering the inverse discrete transform as a cosine series expansion of the original continuous signal, assuming sampling in accordance with the Nyquist criterion. This approach can be applied in the signal domain, to give, for example, DCT based interpolation or derivatives. The same approach can be taken in decoding from the DCT to give, for example, derivatives in the signal domain. The techniques may prove useful in compressed domain processing applications, and are interesting because they allow operations from the continuous domain such as differentiation to be implemented in the discrete domain. An image matching algorithm illustrates the use of the properties, with improvements in computation time and matching quality.

*Key words:* DCT; Image Compression; Derivative; Shift; Scale; Image Matching;

## 1 Introduction

The Discrete Cosine Transform (DCT) [1,2] has found wide application in image and video compression, and continues to be at the centre of innovative research, with recent publications focussing on computation speed (e.g. [3] [4], and transform domain filtering (e.g. [5]). In this paper, a novel interpretation of the DCT is presented, which allows some interesting properties to be derived. These properties include ways to generate the DCT of shifted, scaled or differentiated versions of a signal, directly from its DCT coefficients. The method is based on interpretting the DCT coefficients as the coefficients of a cosine series expansion of a band-limited, symmetrically extended continuous signal. Similar approaches based on splines and polynomial bases have been reported [6,7]. In Section 2 we establish the validity of treating the DCT as a sum of continuous sinusoidal bases. In Section 3, we use this interpretation to derive properties for DCT domain shifting, scaling and differentiation, and in Section 4, we illustrate the use of the properties in an image matching algorithm.

## 2 DCT as Sum of Continuous Basis Functions

Usually, the DCT is interpreted as a sum of discrete bases, summing to a discrete sequence. In this section we interpret the DCT as a sum of continuous

* Corresponding author
*Email address:* r.reeves@qut.edu.au (Robert Reeves).

bases. These bases sum to the symmetrically extended, band-limited continuous signal, which when sampled, gives rise to the discrete sequence referred to above.

Let $g(x)$ be a band limited continuous signal, such that $\omega_{max} < \pi$. Without loss of generality, a sampling interval of one is used, producing $N$ samples $g(n)$ at $n = 0, \ldots, N-1$. The forward discrete transform and its inverse are defined as

$$G(m) = T\{g(n)\} = \sum_{n=0}^{N-1} g(n) f_n(m) \tag{1}$$

and

$$g(n) = T^{-1}\{G(m)\} = \sum_{m=0}^{N-1} G(m) r_m(n) \tag{2}$$

where $f_n(m)$ is the forward transform kernel, $r_m(n)$ is the reverse transform kernel, and $n$ and $m$ are integers from 0 to $N-1$. The type-2 DCT [8], as used in the JPEG [1] and related compression schemes is defined by

$$f_n(m) = r_m(n) = c(m)\frac{\sqrt{2}}{\sqrt{N}} \cos((2n+1)m\pi/2N) \tag{3}$$

with $c(m) = \frac{1}{\sqrt{2}}$ for $m = 0$ and $c(m) = 1$ otherwise, and $g(x)$ is assumed to be symmetrically extended with period $2N$, so that $g(x) = g(x + 2N)$ and $g(-\frac{1}{2} + x) = g(-\frac{1}{2} - x)$.

Though $r_m(n)$ is defined only for integer values of $n$, the expression can be computed for any real value. Replacing the discrete $n$ by real $x$ gives a sum of continuous cosine basis functions,

$$\hat{g}(x) = \sum_{m=0}^{N-1} r_m(x) G(m). \tag{4}$$

By considering the periodicity of the bases $r_m(x)$ and the orthonormality of the DCT kernel, it is evident that when $\hat{g}(x)$ is sampled with an interval of one, the values $g(n)$ and their symmetric and periodic repetitions result, as follows:

$$\hat{g}(x) = \sum_{m=0}^{N-1} r_m(x) \sum_{n=0}^{N-1} g(n) f_n(m) \tag{5}$$

$$= \sum_{n=0}^{N-1} g(n) \sum_{m=0}^{N-1} r_m(x) f_n(m) \tag{6}$$

Taking samples at values of $x = p$, where $p$ is an integer from 0 to $N-1$, and noting that $f_n(m) = r_m(n)$ we have

$$\hat{g}(p) = \sum_{n=0}^{N-1} g(n) \sum_{m=0}^{N-1} r_m(p) f_n(m) \tag{7}$$

$$= \sum_{n=0}^{N-1} g(n) \sum_{m=0}^{N-1} f_p(m) f_n(m). \tag{8}$$

By othonormality of the DCT kernel, $\sum_{m=0}^{N-1} f_p(m) f_n(m)$ is equal to zero unless $p = n$, and one otherwise. Thus sampling $\hat{g}(p)$ produces the same samples $g(n)$ as sampling $g(x)$. This is sufficient to imply the equivalence of these two signals, as long as $g(x)$ is sampled in accordance with the Nyquist criterion. It is trivial to extend this argument to those values of $p$ outside the range 0 to $N-1$ by considering the periodic and symetric extensions of $r_m(p)$: $r_m(p) = r_m(p+2N)$, and $r_m(p) = r_m(-1-p)$.

## 3 DCT Domain Properties

In this section simple expressions are derived for computing the DCT of any linear operation on a signal, from the DCT coefficients of the original signal's samples. Applying a linear operation to both sides of (4), and adopting the

notation $g_L(n)$ to mean the linearly transformed signal sampled at $x = n$, and $r_{Lm}(n)$ to refer to the linearly transformed kernel sampled at $x = n$,

$$g_L(n) = \sum_{m=0}^{N-1} G(m) r_{Lm}(n). \tag{9}$$

It follows from (1) and (9) that

$$T\{g_L(n)\}(m) = \sum_{p=0}^{N-1} G(p) \sum_{n=0}^{N-1} f_n(m) r_{Lp}(n). \tag{10}$$

This represents a linear transform which computes the DCT of $g_L(n)$ from the DCT of $g(n)$, in a single matrix multiplication. The values of the terms $\sum_{n=0}^{N-1} f_n(m) r_{Lp}(n)$ are independent of the signal and its samples, depending only on the type of linear transformation.

Differentiation is one example of a linear property that can be performed in the DCT domain. In this case $g_L(n)$ denotes the derivative of $g(x)$ sampled at $x = n$, and

$$r_{Lp}(n) = -c(p) \frac{\sqrt{2}}{\sqrt{N}} \frac{m\pi}{N} \sin((2n+1)p\pi/2N), \tag{11}$$

given by the derivative of the reverse transform kernel, sampled at $x = n$. The extension to second and higher derivatives, real valued shift, scaling, integrals, or any combination of them, is trivial. For example, a shifting and scaling property is given by letting $r_{Lp}(n) = r_p(a_0 + a_1 n)$.

## 4 Example - image matching

As an example of how these DCT properties can be used, two dimensional versions were incorporated into an image matching algorithm based on the

standard approach of Ackerman [9], in which partial derivatives are required. An affine transformation models the transformation of left image patch to right image patch as follows,

$$g_1(x,y) = h_0 + h_1 g(a_0 + a_1 x + a_2 y, b_0 + b_1 x + b_2 y) + n_1(x,y) \tag{12}$$

and

$$g_2(x,y) = g(x,y) + n_2(x,y) \tag{13}$$

where $g_1(x,y)$ and $g_2(x,y)$ are the image patches to be matched, $h_0$ and $h_1$ are radiometric transformation parameters, $a_i$ and $b_i$ are geometric transformation parameters, and $n_1(x,y)$ and $n_2(x,y)$ are Gaussian noise.

Using Taylor's theorem to linearize each equation about an initial guess and then subtracting yields

$$\begin{aligned}\Delta g(x,y) = & dh_0 + dh_1 g(x,y) + da_0 \frac{\partial}{\partial x} g(x,y) + da_1 x \frac{\partial}{\partial x} g(x,y) \\ & + da_2 y \frac{\partial}{\partial x} g(x,y) + db_0 \frac{\partial}{\partial y} g(x,y) \\ & + db_1 x \frac{\partial}{\partial y} g(x,y) + db_2 y \frac{\partial}{\partial y} g(x,y) + v(x,y)\end{aligned} \tag{14}$$

where $x$ and $y$ take on a series of discrete values within a match window. This results in a system of equations for the perturbations to the initial radiometric and geometric transformation parameters.

The system of equations can be expressed in matrix form

$$\boldsymbol{L} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{v} \tag{15}$$

with the solution given by

$$\hat{\boldsymbol{x}} = (\boldsymbol{A}^T\boldsymbol{A})^{-1}\boldsymbol{A}^T\boldsymbol{L} \tag{16}$$

where $\hat{\boldsymbol{x}}$ is the vector of perturbations to the initially chosen transformation parameters that result in a better match between the two image patches. Vector $\boldsymbol{v}$ is a vector of noise terms, and $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & g(\cdot) & \frac{\partial}{\partial x}g(\cdot) & x\frac{\partial}{\partial x}g(\cdot) & y\frac{\partial}{\partial x}g(\cdot) & \frac{\partial}{\partial y}g(\cdot) & x\frac{\partial}{\partial y}g(\cdot) & y\frac{\partial}{\partial y}g(\cdot) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{17}$$

Since the solution is based around a linear approximation, it can be improved by linearizing around the new solution, and re-solving. This is repeated until the solution converges.

By choosing a suitable ordering system, the images can be expressed as column vectors, the 2D linear transform as a matrix, and (15) can be expressed in the transform domain as,

$$\boldsymbol{TAx} + \boldsymbol{Tv} = \boldsymbol{TL} \tag{18}$$

where multiplying by matrix $\boldsymbol{T}$ takes the 2D DCT transform. This can be viewed as defining transform domain $\boldsymbol{A}$ and $\boldsymbol{L}$ matrices given by $\boldsymbol{TA}$ and $\boldsymbol{TL}$. It has been shown previously that as long as $\boldsymbol{T}$ is orthogonal, which is the case for the DCT, the solution of (16) is unaffected by using the transform domain $\boldsymbol{A}$ and $\boldsymbol{L}$ matrices [10]. For typical images, the DCT behaves in a similar manner to the Karhunen-Loeve transform, which constructs basis functions

in order of decreasing variance. In image compression, this fact is used to justify discarding many of the high frequency (low variance) coefficients, while maintaining the information important to the structure of the image [11]. This same principle can be extended to image matching. Since the bulk of the image energy appears in the low order DCT coefficients, discarding the higher order coefficients should not impair image matching. We can significantly reduce the size of the $\boldsymbol{A}$ matrix by transforming each column into the DCT domain, and then omitting the same high frequency coefficients from each column. Since the computational effort in the solution of the least squares system depends on the size of matrix $\boldsymbol{A}^T\boldsymbol{A}$, this should enable the solution to be computed more quickly, without detriment to the quality of the match result. The method of (10) is used to compute the transforms of the columns of the $\boldsymbol{A}$ matrix involving partial derivatives, from the transform of the image patch. The extension to two dimensions is straightforward, with full details given in [12] An experimental investigation is fully reported elsewhere [12,13]. Here we briefly summarise the main results concerning accuracy and computation time, when compared to a fully pixel domain algorithm in which the partial derivaives are estimated by first differences.

It is important to note that as far as this application is concerned, the important point that results in computational efficiencies is that the least squares problem (18) is solved in the transform domain after removing those transform domain equations which effectively involve only noise. While we have found it expedient to use the transform domain properties we have proposed to compute the transforms of the rows of $\boldsymbol{A}$ which involve partial derivatives from the transform of the image patch, an equivalent procedure would be to first compute the collumns of $\boldsymbol{A}$, finding the partial derivatives by some other

means, and then taking the DCT of each collumn of $\boldsymbol{A}$. However this introduces the problem of estimating the partial derivatives. Apart from the issues of the assumed peiodic extension, and the satisfaction of the Nyquist criterion, the partial derivatives involved in the methods we propose are those of the original continuous function, not discrete estimates with an associated imprecision. As we discuss in Section 4.3 analagous properties in the time(space) domain can be used to estimate the collumns of $\boldsymbol{A}$. However in this case we would have an additional DCT to perform for each collumn of $\boldsymbol{A}$ involving a partial derivative.

An artificial horizontal disparity was introduced into two fragments of aerial photographs as follows. In image one (figure 1), the left image was formed by subsampling a $1600 \times 1600$ fragment using a $10 \times 10$ Gaussian window, while the right image was formed by first shifting by 5 pixels, then subsampling. In image two (figure 2), the left image was formed by subsampling a $328 \times 328$ fragment using a $2 \times 2$ Gaussian window, while the right image was formed by first shifting by 1 pixel, then subsampling. This resulted in a known disparity of 0.5 pixels being introduced in each case between the left and right images.

[Insert Figures 1 and 2]

### *4.1 Standard Deviation of the Disparity Errors*

The standard deviation of the disparity errors represents the accuracy of the matching result, and in the case of the errors having zero mean, which is approximately true for our data, it is equivalent to the RMS error in the disparity estimate. Figure 3 shows that for all window sizes, and for two different aerial images, the standard deviation of the errors starts off large, and as more

DCT coefficients are added, quickly reduces, and then flattens out, after which adding further DCT coefficients has little impact on the accuracy. The knee in the curve occurred at around 5% to 10% of the DCT coefficients for the $32 \times 32$ window, at around 20% for the $16 \times 16$ window, and around 25% to 30% for the $8 \times 8$ window. In all cases, after the knee, the accuracy was comparable or better than that achieved by the pixel domain algorithm, sometimes markedly so.

[Insert Figure 3 ]

### *4.2 Average Convergence Time*

The average time for match windows to converge is shown in Figure 4. For the $8 \times 8$ window, the times for the DCT domain algorithm are comparable to the pixel domain for DCT coefficient percentages up to about 30%, but then gradually increase as further DCT coefficients are added. For the $16 \times 16$ window, taking between 10% and 30% of the coefficients resulted in reducing the average convergence time to about 50% of the pixel domain time in one image, and about 75% in the other image. The improvements were more pronounced for the $32 \times 32$ window, where in both images the average convergence time was under 50% of the pixel domain time, when between 5% and 20% of the DCT coefficients were taken.

[Insert Figure 4]

From our use of these DCT domain properties, several important considerations emerge. Firstly, the properties are based on the assumption that the signal is symmetrically extended at each end of the DCT window [14], or block in the 2D case. Where the result of the linear operation is outside the DCT window (as possible with shifting or scaling), a point on the symmetrically extended waveform is returned. In the case of the DCT support being the entire signal or image, this may be an acceptable edge effect. However in block based decompositions, edge effects may be introduced into each block. The symmetric extension also causes the derivative to tend towards zero at the edges of the DCT window. This may also be problematic in block based schemas.

Time (space) domain versions of the property provide a means of shifting, scaling and taking the derivative of sampled signals. For example, a shifted signal can be computed by

$$g(n+a) = \sum_{n=0}^{N-1} g(n) \sum_{m=0}^{N-1} f_n(m) r_m(n+a). \tag{19}$$

This equation represents a linear transform, based on the DCT kernels and the shift parameter. Note that the shift parameter $a$ can be any real value. This equation can also be viewed as an interpolation function. Such an interpolation can be combined with scaling and taking the derivative. It differs from a DCT interpolation technique proposed by Wang [15–17] which results in an increased number of samples, spanning the same signal support. In the method proposed here, the number of samples remains fixed, but the signal support may change if scaling or shifting is involved.

The property can also be incorporated directly into the decoding step by making use of (9).

## 5 Summary and Conclusions

Shift, scale and derivative properties for the DCT can be derived by treating the inverse transform as a sum of continuous cosine bases. This sum of continuous bases is identical to the original continuous signal, subject to the Nyquist criterion and the assumed symmetric periodic extension.

A single linear transform can be used to compute the DCT of the sampled derivative, from the DCT of the original signal. Linear transforms can also be constructed for other linear operations, for example shifting and scaling. Any number of sequentially applied linear operations may be combined into a single linear transform, based on applying the combined transform to the cosine bases.

The property described in this paper may also be applied in the time or space domain, for example, to shift a signal by a real (possibly fractional) number of samples, or to differentiate it. Given that the 2D DCT is separable, there is no impediment to a straightforward 2D extension, which has been used in a DCT domain image matching algorithm. We expect therefore that these techniques may be useful for DCT based image representations, particularly where geometric transformations or derivatives are required.

As an example of how these properties may be used, a standard least squares image matching algorithm was implemented in the DCT domain, making use of the properties described. The algorithm was able to perform more accu-

rately and converge faster than a comparable pixel domain algorithm using first differences to estimate the partial derivatives. This improved performance may be attributed to two factors. Firstly, the DCT domain algorithm enables us to discard a high percentage of DCT coefficients in the least squares adjustment, thus reducing the size of the solution without losing significant image information. Secondly, the DCT properties described provide a better estimate of the partial derivatives than the first differences.

## References


[1] G. K. Wallace, The JPEG still-picture compression standard, Communications of the ACM 34 (4) (1991) 31–44.

[2] K.-H. Tzou, Video coding techniques: An overview, in: P. Pirsch (Ed.), VLSI Implementations for Image Communications, Elsevier Science Publishers, 1993, pp. 1–47.

[3] S. Lee, Improved algorithm for efficient computation of the forward and backward MDCT in MPEG audio coder, IEEE Transactions on Circuits and Systems II-Analog and Digital Signal Processing 48 (10) (2001) 990–994.

[4] J. Liang, T. Tran, Fast multiplierless approximations of the DCT with the lifting scheme, IEEE Transactions on Signal Processing 49 (12) (2001) 3032–3044.

[5] N. Nikolaev, A. Gotchev, K. Egiazarian, Z. Nikolov, Suppression of electromyogram interference on the electrocardiogram by transform domain denoising., Medical and Biological Engineering and Computing 39 (6) (2001) 649–655.

[6] M. Unser, Splines - a perfect fit for signal and image processing, IEEE Signal Processing Magazine 16 (6) (1999) 22–38.

[7] H. Ridha, J. Vesma, T. Saramaki, M. Renfors, Derivative approximations for sampled signals based on polynomial interpolation, in: Proceedings of the 13th International Conference on Digital Signal Processing, Vol. 2, IEEE, 1997, pp. 939–942.

[8] K. Rao, P. Yip, Discrete Cosine Transform - Algorithms, Advantages, Applications, Academic Publishers, San Diego, 1990.

[9] F. Ackermann, Digital image correlation: Performance and potential application in photogrammetry, Photogrammetric Record 11 (64) (1984) 429–439.

[10] R. Reeves, K. Kubik, Least squares matching in the transform domain, International Archives of Photogrammetry and Remote Sensing 32 (3/1) (1998) 168–176.

[11] M. Rabbini, P. Jones, Digital Image Compression Techniques, SPIE Optical Engineering Press, Bellingham, WA, 1991.

[12] R. Reeves, Image matching in the compressed domain, Ph.D. thesis, Space Centre for Satellite Navigation, Queensland University of Technology, Brisbane, Australia (1999).

[13] R. Reeves, K. Kubik, Benefits of hybrid DCT domain image matching., International Archives of Photogrammetry and Remote Sensing 32 (2000) 761–768.

[14] S. A. Martucci, Symmetric convolution and the discrete sine and cosine transforms, IEEE Transactions on Signal Processing 42 (5) (1994) 1038–1051.

[15] Z. Wang, Interpolation using type I discrete cosine transform, Electronics Letters 26 (15) (1990) 1170–1171.

[16] Z. Wang, Interpolation using the discrete cosine transform: reconsideration, Electronics Letters 29 (2) (1993) 198–200.

[17] J. Agbinya, Two dimensional interpolation of real sequences using the DCT, Electronics Letters 29 (2) (1993) 204–205.

## 6 Figures

Fig. 1. *Image one was formed by subsampling a 1600 × 1600 fragment using a 10 × 10 Gaussian window, while the right image was formed by first shifting by 5 pixels, then subsampling.*

Fig. 2. *Image two was formed by subsampling a 328 × 328 fragment using a 2 × 2 Gaussian window, while the right image was formed by first shifting by 1 pixel, then subsampling.*

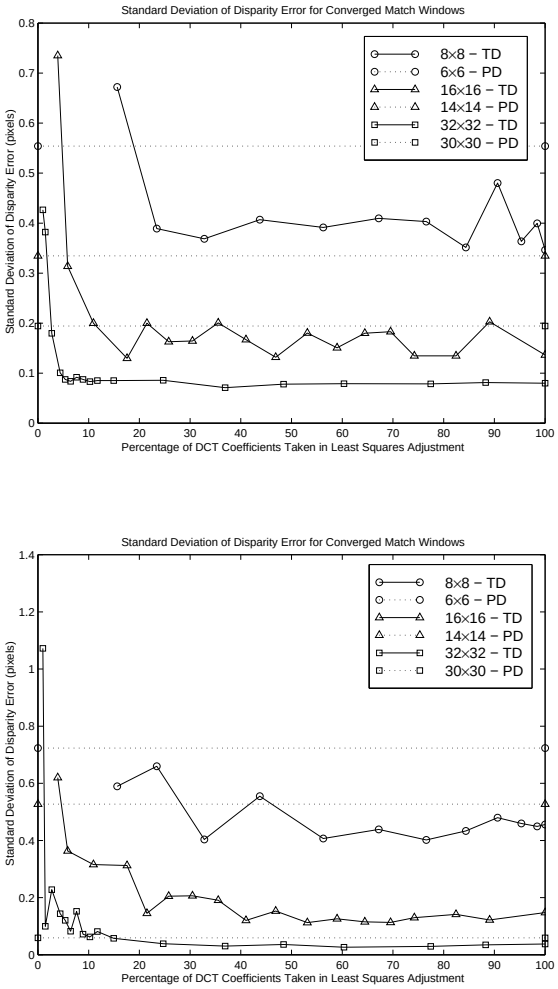


Fig. 3. *Optimum matching accuracy is shown to be acchieved with a small percentage of the available DCT coeeficients. Results are shown for* $8 \times 8$, $16 \times 16$ *and* $32 \times 32$ *windows for image one (top) and image two (bottom). The accuracy acchieved by a comparable pixel domain algorithm are shown as dotted lines for each of three window sizes.*

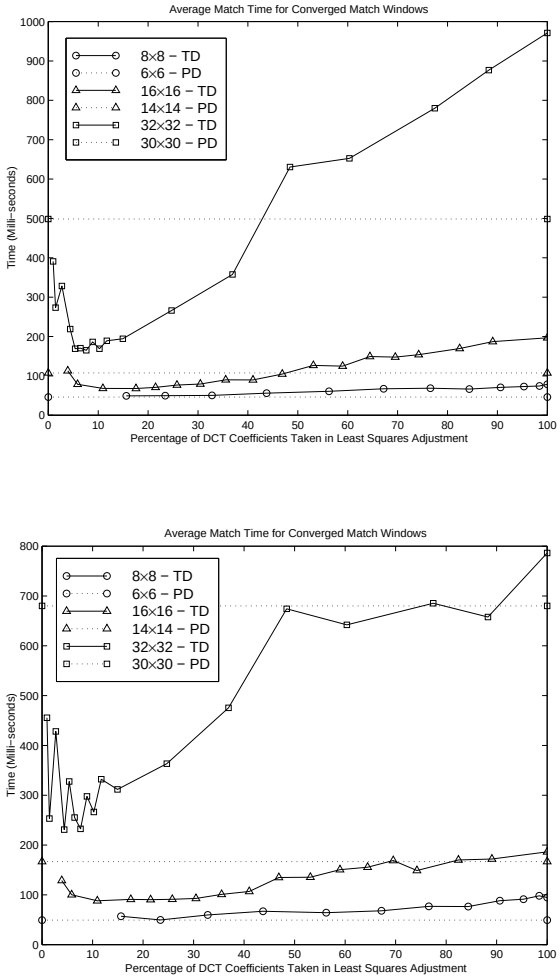


Fig. 4. *The effect of taking only a fraction of the available DCT coefficients in each least squares adjustment on the average time taken to converge for each match window is shown for image one (top) and image two (bottom). The times for the pixel domain algorithms are shown as dotted horizontal lines for comparison.*